Scheduling to minimize over-regular criteria

Chengbin Chu

UNITÉ DE RECHERCHE
INRIA-LORRAINE

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

1992
25ème anniversaire

N° 1704

Programme 5
Traitement du Signal,
Automatique et Productique

SCHEDULING TO MINIMIZE OVER-REGULAR CRITERIA

Chengbin CHU

Juin 1992

RR-1704

SCHEDULING TO MINIMIZE OVER-REGULAR CRITERIA

PROBLEMES D'ORDONNANCEMENT POUR MINIMISER DES CRITERES "PLUS QUE REGULIERS"

Chengbin Chu

Projet SAGEP, INRIA-Lorraine, CESCOM Technopôle Metz 2000,
4 rue Marconi, 57070 Metz, France

Abstract. In this paper, we define a new class of scheduling criteria called over-regular criteria. This class is a subclass of regular criteria. We analyze the properties of scheduling problems with over-regular criteria. In the literature a lot of effort has been made to solve specific problems, these properties, however, are applicable to several scheduling problems. Using these properties and proving additional specific ones, we propose a procedure to solve the scheduling problem to minimize total tardiness with generalized due dates. This latter problem particularly arises in maintenance departments where maintained parts are interchangeable. Computational results are also reported.

Résumé. Dans cet article, nous définissons une nouvelle classe de critères d'ordonnancement appelés critères "over-regular" ou "plus que réguliers". Cette classe est un sous ensemble de la classe de critères réguliers. Nous analysons les propriétés des problèmes d'ordonnancement avec critères "plus que réguliers". Par rapport à la littérature où beaucoup d'effort a été apporté à la résolution des problèmes spécifiques, nos résultats sont applicables à plusieurs problèmes d'ordonnancement. En utilisant ces propriétés et en démontrant des propriétés spécifiques supplémentaires, nous proposons une procédure de résolution pour minimiser la somme des retards avec délais généralisés. Ce dernier problème se pose en particulier dans les services de maintenance où les pièces réparées sont interchangeables. Des résultats numériques sont également fournis.

1. INTRODUCTION

In the scheduling literature, a lot of effort has been made to solve specific problems. If some context changes, the results become inapplicable. General results are limited to the definition of regular criteria and the dominance of semi-active and active schedules (Conway, Maxwell, and Miller 1967, Baker 1974). In this paper, we define a new class of scheduling criteria called over-regular criteria. We prove a number of dominance properties applicable for any one of the criteria in the case of one-machine scheduling. We also show that these criteria can be minimized by scheduling the jobs according to the shortest remaining processing time (SRPT) priority rule if the jobs are pre-emptive. Using these properties and proving supplementary dominance properties, we propose some branch and bound algorithms to solve a particular over-regular criterion scheduling problem: minimizing total tardiness with generalized due dates.

The concept of generalized due dates has been introduced by Hall (1986). Traditionally, the due dates are job specific, that is, with each job i is associated a due date d_i. If job i is completed after its due date d_i, it is said to be tardy. With generalized due dates, however, the due dates are independent of jobs. If we denote by $d_{[j]}$ the jth smallest due date (i.e. $d_{[1]} \leq d_{[2]} \leq \ldots \leq d_{[n]}$, where n is the number of jobs), the job completed at the jth position will be considered to be tardy if its completion time $C_{[j]}(S)$ in a schedule S is greater than $d_{[j]}$. In the same paper, Hall also described a number of applications in which the generalized due date definition is appropriate. These applications include publicity planning problems, survey design, and scheduling problems in some manufacturing environments. Hall, Sethi and Sriskandarajah (1991) cited a particular instance in the petrochemical industry, where a number of interchangeable heat exchangers must be maintained. The repair time of a heat exchanger is estimated with respect to the extent of corrision and erosion on it. Since a number of heat exchangers must be serviced by a certain date, the problem can be formulated as a generalized due date scheduling problem.

This paper considers scheduling problems with over-regular criteria minimization involving generalized due dates. Section 2 is devoted to the definition of over-regular criteria and related analysis. Section 3 considers the total tardiness minimization with generalized due dates.

2. OVER-REGULAR CRITERIA AND THEIR PROPERTIES

In the literature, a regular criterion is defined as follows (Conway, Maxwell, and Miller 1967, Baker 1974).

Definition 1. Let $C_i(S)$, $G(S)$ denote, respectively, the completion time of job i and the value of criterion G in a schedule S. If $C_i(\sigma) \leq C_i(\sigma')$ $\forall i=1, 2, \ldots, n$ implies $G(\sigma) \leq G(\sigma')$ for two schedules σ and σ', the criterion G is said to be a regular criterion.

According to the notations, it is not difficult to see that the series $(C_{[1]}(S), C_{[2]}(S) \ldots, C_{[n]}(S))$ is obtained by reordering in nondecreasing order the series $(C_1(S), C_2(S), \ldots, C_n(S))$.

Other notions already defined in the literature (Baker 1974 for example) are dominant subset and active schedules. A *dominant subset* is a subset of solutions containing at least one optimal solution. An *active schedule* is such that no job can be scheduled earlier without delaying another. It has been shown that the subset of active schedules is dominant for regular criteria (Baker 1974).

Now we define a new class of criteria called over-regular as follows.

Definition 2. If for any two schedules σ and σ', $C_{[j]}(\sigma) \leq C_{[j]}(\sigma')$ $\forall j=1, 2, \ldots, n$ implies $G(\sigma) \leq G(\sigma')$, G is said to be an over-regular criterion.

In fact, if for two schedules σ and σ' we have $C_i(\sigma) \leq C_i(\sigma')$, $\forall i=1, 2, \ldots, n$, from the definition, it is easy to see that $C_{[j]}(\sigma) \leq C_{[j]}(\sigma')$ $\forall j=1, 2, \ldots, n$. If G is an over-regular criterion, then $G(\sigma) \leq G(\sigma')$, Consequently, the following property holds.

Property 1. Any over-regular criterion also is a regular criterion.

It is from this property that we use the name "over-regular" criterion. Let us look at some scheduling criteria.

(a) Makespan $C_{\max}(S)=\max_{i=1,2,\ldots,n}\{C_i(S)\}=C_{[n]}(S)$.

(b) Total completion time $C(S)=\sum_{i=1}^{n} C_i(S) =\sum_{j=1}^{n} C_{[j]}(S)$.

(c) Total tardiness with generalized due dates $Q(S)=\sum_{j=1}^{n} Q_{[j]}(S)$, where

$Q_{[j]}(S)=\max\{C_{[j]}(S)-d_{[j]},0\}$.

(d) Maximum lateness with generalized due dates $R_{\max}(S)=\max_{j=1,2,\ldots,n}\{R_{[j]}(S)\}$, where

$R_{[j]}(S)=C_{[j]}(S)-d_{[j]}$.

(e) Total number of tardy jobs with generalized due dates $U(S)=\sum_{j=1}^{n} U_{[j]}(S)$, where

$$U_{[j]}(S)=\begin{cases}1 & \text{if } C_{[j]}(S) > d_{[j]},\\ 0 & \text{otherwise.}\end{cases}$$

It is easy to see that all these criteria are over-regular. Since the total completion time minimization is equivalent to total (mean) flow time, total (mean) waiting time and total (mean) lateness minimization (Conway, Maxwell, and Miller 1967), these latter criteria are over-regular as well. In single machine case, these problems can be efficiently solved by an algorithm proposed by Chu (1992b).

In the remainder of this section, we establish some properties for over-regular criteria in the case of single machine scheduling. In this problem, a set of jobs $N=\{1, 2, \ldots, n\}$ have to be scheduled on a single machine assumed to be able to process at most one job at a time. Each job i has a release date r_i at which it becomes available to be processed, and a processing time p_i. Once a job starts being processed, the processing can not be interrupted.

The following theorem show that if the jobs are pre-emptive, the over-regular criteria minimization problem can be polynomially solved by applying the shortest remaining processing time (SRPT) priority rule. According to this rule, jobs are scheduled portion after portion from time $-\infty$. When a job is completed, a waiting job with the shortest remaining processing time is chosen to be processed by the machine. When a new job arises, it takes possession of the machine if its processing time is strictly smaller than the remaining processing time of the job in process.

Theorem 1. The single machine scheduling problem to minimize over-regular criteria with pre-emptive jobs is polynomially solvable by applying the SRPT rule. The complexity of the problem is $O(n\log n)$.

Proof. Let σ be a schedule obtained by applying the SRPT rule and σ' *any* feasible schedule. We proved (Chu 1992a) that $C_{[j]}(\sigma)\le C_{[j]}(\sigma')$, $\forall j=1, 2, \ldots, n$. From the definition of over-regular criterion G, we have $G(\sigma)\le G(\sigma')$. This means that σ is an optimal schedule. In the same paper, we also showed that the SRPT rule can be implemented in $O(n\log n)$.

From now on, we prove some dominance properties for over-regular criteria minimization. Before doing this, we define what "dominance" and "dominance property" mean.

Definition 3. Consider a scheduling problem where the criterion G is to be minimized. If two schedules σ and σ' are such that $G(\sigma)\leq G(\sigma')$, we say that schedule σ dominates schedule σ'. A dominance property indicates the conditions under which a schedule dominates another.

According to the definition, we can see that dominance properties are very useful to reduce the search for an optimal solution and hence to reduce computational effort because we do not need to consider dominated solutions.

From the definition of over-regular criteria, in order to prove that σ dominates σ', it is sufficient to prove that $C_{[j]}(\sigma)\leq C_{[j]}(\sigma')$, $\forall j=1, 2, \ldots, n$. Another way to prove this is to use the following property.

Property 2. If two schedules σ and σ' and two jobs i and k are such that

(i) $\min\{C_i(\sigma),C_k(\sigma)\} \leq \min\{C_i(\sigma'),C_k(\sigma')\}$,

(ii) $\max\{C_i(\sigma),C_k(\sigma)\} \leq \max\{C_i(\sigma'),C_k(\sigma')\}$ and

(iii) $C_l(\sigma)\leq C_l(\sigma')$, $\forall l$, $l\neq i$ and $l\neq k$,

then σ dominates σ' for any over-regular criterion.

Proof. The conditions described in Property 2 imply that $C_{[j]}(\sigma)\leq C_{[j]}(\sigma')$, $\forall j=1, 2, \ldots, n$.

In the following we use this property or directly the definition to prove dominance properties, where $R_i(t)=\max(t,r_i)$ and $F_i(t)=R_i(t)+p_i$ are, respectively, the earliest start time and the earliest finish time of job i at time t.

Theorem 2. Let $j\in N$ be a job such that $r_j=\max_{i\in N}(r_i)$, there is an optimal schedule in which job j is preceded by any job k such that $p_k\leq p_j$.

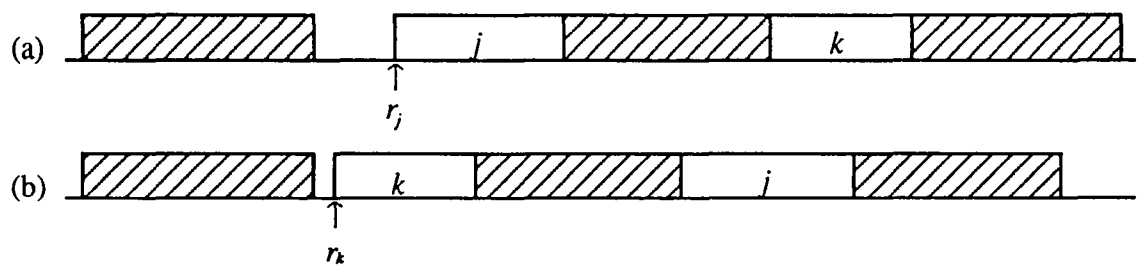

Figure 1. Interchange of positions of jobs j and k. (a) Schedule S. (b) Schedule S'.

Proof. Consider a schedule S such that job j precedes job k. We construct a new schedule S' by interchanging the positions of jobs j and k (see Figure 1). It is clear from the figure, that the other jobs than j and k are finished earlier than in S, and that $C_j(S') \leq C_k(S)$, $C_k(S') \leq C_j(S)$.

Before proceeding with the presentation, we introduce some notations. Let t_i denote the completion time of the job preceding immediately job i in a given schedule. If i is the first job in the schedule, t_i is, by convention, set to $-\infty$.

Theorem 3. If there is a job i such that $i \in N$, and $\forall j \in N$, $p_i \leq p_j$ (i.e. i is a job with the shortest processing time), there is an optimal schedule in which job i precedes any job k such that $r_i \leq r_k$.

Proof. Let us consider a schedule S in which jobs i and job k satisfy the assumptions of Theorem 3 and in which job k precedes job i. Let us denote by k' the job immediately preceding i. Job k' is then either job k, or between jobs k and i. In any case, we have $r_i \leq r_k \leq R_{k'}(t_{k'})$ and therefore $R_i(t_{k'}) \leq R_{k'}(t_{k'})$. Taking into account the fact that $p_i \leq p_{k'}$, we have $F_i(t_{k'}) \leq F_{k'}(t_{k'})$. We then interchange positions of jobs k' and i without delaying any other jobs to obtain a new schedule $S^{(1)}$ (see Figure 2). The inequality $R_i(t_{k'}) \leq R_{k'}(t_{k'})$ implies that $S^{(1)}$ is feasible and $C_{k'}(S^{(1)}) \leq C_i(S)$. The inequality $F_i(t_{k'}) \leq F_{k'}(t_{k'})$ implies that $C_i(S^{(1)}) \leq C_{k'}(S)$. Consequently, $G(S^{(1)}) \leq G(S)$. This process can be continued until a schedule $S^{(m)}$ is obtained in which job k follows job i. The same reasoning can be applied to obtain $G(S^{(m)}) \leq \ldots G(S^{(1)}) \leq G(S)$.

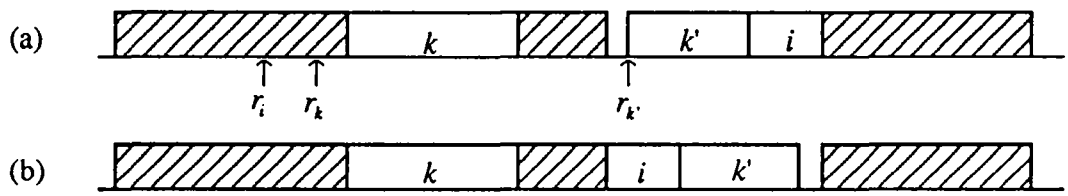

Figure 2. Interchange of positions of jobs i and k'. (a) Schedule S. (b) Schedule $S^{(1)}$.

Henceforth, let K denote a partial schedule, $J(K)$ the set of jobs in this partial schedule, $q(K)$ the number of jobs in this partial schedule and $E(K)$ the completion time of the last job in K. K/i is the new partial schedule obtained by adding the job i behind the given partial schedule K. $P(K,i)$ is the schedule composed of K/i, completed by the partial optimal schedule of jobs belonging to N-$J(K/i)$ starting from the moment $E(K/i)=F_i(E(K))$.

Theorem 4. If $\max_{i\in N-J(K)}(r_i) \le \min_{i\in N-J(K)}\{F_i(E(K))\}$, and j and k are two jobs belonging to N-$J(K)$ such that $R_j(E(K))\le R_k(E(K))$ and $F_j(E(K))\le F_k(E(K))$ then $P(K,j)$ dominates $P(K,k)$.

Proof. Consider the schedule $P(K,k)$. Construct another schedule S by exchanging the positions of j and k (see Figure 3). From the figure, it is clear that the other jobs can be processed earlier in S than in $P(K,k)$, because $\max_{i\in N-J(K)}(r_i) \le \min_{i\in N-J(K)}\{F_i(E(K))\}$, $R_j(E(K))\le R_k(E(K))$ and $F_j(E(K))\le F_k(E(K))$. Furthermore, we have $C_j(S)\le C_k(P(K,k))$ and $C_k(S)\le C_j(P(K,k))$.

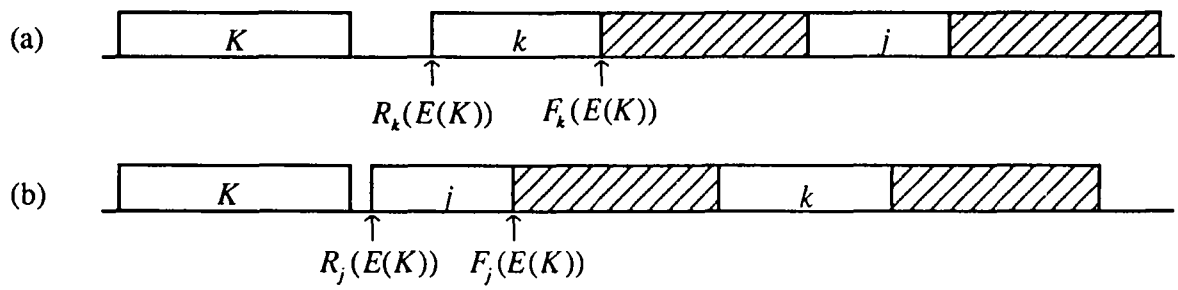

Figure 3. Interchange of positions of jobs j and k. (a) Schedule $P(K,k)$. (b) Schedule S.

Theorem 5. Given two jobs $i, j\in N$-$J(K)$, if $p_i\ge p_j$ and $F_i(E(K))\le F_j(E(K))$, then $P(K,i)$ dominates $P(K,j)$.

Proof. Let us consider the schedule $P(K,j)$. We then construct another schedule S, by exchanging the positions of i and j and the positions of other jobs remain unchanged (see Figure 4). This schedule is feasible because $F_i(E(K))\le F_j(E(K))$ and $p_i\ge p_j$. These conditions imply furthermore $C_i(S)\le C_j(P(K,j))$ and $C_j(S)\le C_i(P(K,j))$.

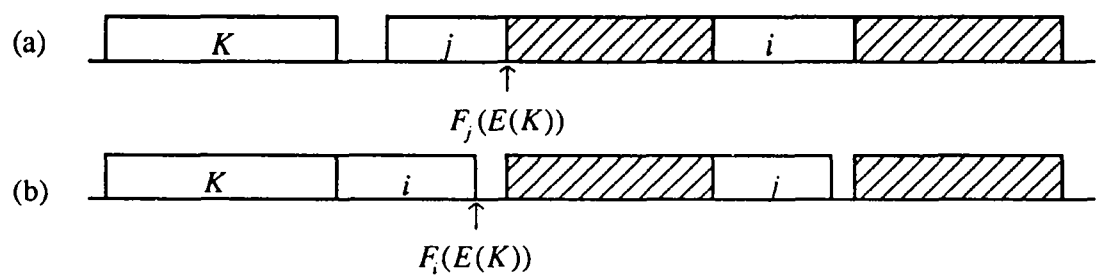

Figure 4. Interchange of positions of jobs i and j. (a) Schedule $P(K,j)$. (b) Schedule S.

Theorem 6. Given two jobs $i, j \in N\text{-}J(K)$, if $F_i(E(K)) \le R_j(E(K)) + \min_{k \in N-J(K)}\{p_k\}$, then $P(K,i)$ dominates $P(K,j)$.

Proof. Consider the schedule $P(K,j)$, then construct another schedule S by scheduling job i before job j and by delaying by $F_i(E(K))$-$R_j(E(K))$ all the jobs between jobs i and j, and the positions of all the jobs after job i remaining the same (See Figure 5).

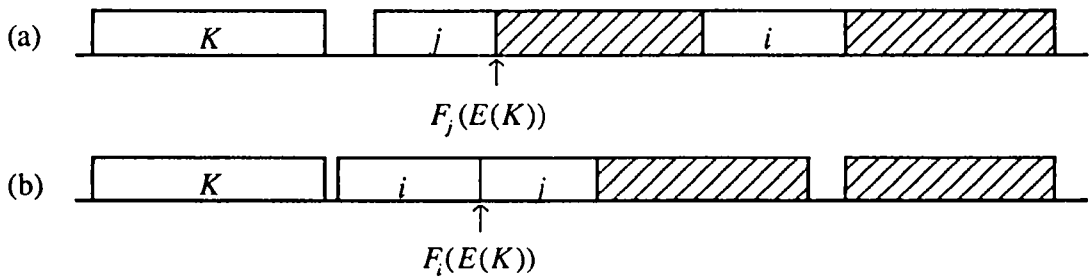

Figure 5. Illustration of Theorem 6. (a) Schedule $P(K,j)$. (b) Schedule S.

Considering the assumption of Theorem 6, we have:

$$R_i(E(K)) = F_i(E(K)) - p_i \le F_i(E(K)) - \min_{k \in N-J(K)}\{p_k\} \le R_j(E(K)).$$

Thus the schedule S is feasible. Suppose that job i is at the πth position in $P(K,j)$. From the construction of schedule S, we have:

(i) $C_{[q(K)+1]}(S)=F_i(E(K))$, $C_{[\pi]}(P(K,j))=C_{[\pi-1]}(P(K,j))+p_i$ and $C_{[q(K)+1]}(P(K,j))=F_j(E(K))$;
(ii) $C_{[l]}(S)=C_{[l-1]}(P(K,j))+F_i(E(K))-R_j(E(K))$, $\forall l=q(K)+2, q(K)+3, \ldots, \pi$.

From the assumption of Theorem 6 and (i), we have

$$F_i(E(K)) \le R_j(E(K)) + \min_{k \in N-J(K)}\{p_k\} \le R_j(E(K)) + p_j = F_j(E(K)),$$

i.e.
$$C_{[q(K)+1]}(S) \le C_{[q(K)+1]}(P(K,j)).$$

On the other hand, it is easy to see that for any feasible schedule σ and for any $l>1$, we should have $C_{[l]}(\sigma) \ge C_{[l-1]}(\sigma) + p_{[l]}$. From (ii) and the assumption of Theorem 6, we have:

$$C_{[l]}(S) \le C_{[l]}(P(K,j)) - p_{[l]} + F_i(E(K)) - R_j(E(K)) \le C_{[l]}(P(K,j)) - \min_{k \in N-J(K)}\{p_k\}$$
$$+F_i(E(K)) - R_j(E(K)) \le C_{[l]}(P(K,j)), \quad \forall l = q(K)+2, q(K)+3, \ldots, \pi.$$

Finally $C_{[l]}(S) \le C_{[l]}(P(K,j)), \quad \forall l = q(K)+1, q(K)+2, \ldots, \pi.$

It should be noticed that Theorem 6 makes the dominance of active schedules redundant, because according to dominance of active schedules, a schedule $P(K,i)$ is dominated if $\exists j \in N\text{-}J(K)$, such that $F_i(E(K)) \le R_j(E(K))$. This condition is stricter than $\exists j \in N\text{-}J(K)$ such that $F_i(E(K)) \le R_j(E(K)) + \min_{k \in N-J(K)}\{p_k\}$.

Theorem 7. There is an optimal schedule such that no pair of adjacent jobs i and i' (i followed by i') are such that $R_{i'}(t_i) \le R_i(t_i)$ and $F_{i'}(t_i) \le F_i(t_i)$, with at least one inequality being strict.

Proof. Suppose that there is an optimal schedule S such that there are two adjacent jobs i and i' (i followed by i') verifying the conditions described in Theorem 7. We construct another schedule S' by inverting the positions of jobs i and i', and the positions of other jobs remaining the same (see Figure 6). Since $R_{i'}(t_i) \le R_i(t_i)$, S' is feasible and $C_i(S') \le C_{i'}(S)$. Since $F_{i'}(t_i) \le F_i(t_i)$, we have $C_{i'}(S') \le C_i(S)$. Therefore, $C_{[j]}(S') \le C_{[j]}(S)$, $\forall j = 1, 2, \ldots, n$. Consequently, $G(S') \le G(S)$. This means that either S is not optimal ($G(S') < G(S)$) which would contradict the assumption that S is an optimal schedule, or S' is also an optimal schedule ($G(S') = G(S)$).

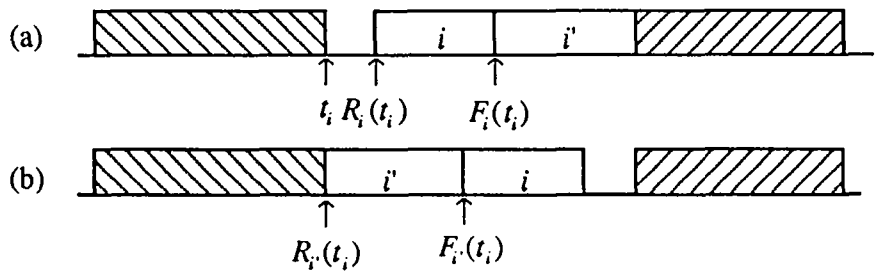

Figure 6. Interchange of positions of jobs i and i'. (a) Schedule S. (b) Schedule S'.

3. MINIMIZING TOTAL TARDINESS WITH GENERALIZED DUE DATES

This section considers the problem of minimizing total tardiness with release dates and generalized due dates. It has been shown to be NP-hard (Hall, Sethi and Sriskandarajah 1991),

but is polynomially solvable by the shortest processing time (SPT) policy, if the release dates are identical.

In the literature, the minimization of total tardiness with respect to specific due dates has received a lot of attention, especially with identical release dates (Baker and Bertrand 1982, Du and Leung 1990, Emmons 1969, Fisher 76, Lawler 1977, 1982, Potts and Van Wassenhove 1982, Srinivasan 1971, Wilkerson and Irwin 1971), among which the algorithm proposed by Potts and Van Wassenhove can solve problems with up to 100 jobs using a decomposition approach and the algorithm of Baker and Schrage (1978) for the problem with precedence constraints between jobs. With different release dates but still with specific due dates, we proved a sufficient condition for local optimality which can also be considered as a dynamic priority rule. On the basis of this priority rule, we proposed some efficient heuristics (Chu and Portmann 1992). We also developed a branch-and-bound algorithm capable of optimally solving hard problems with up to 30 jobs and relatively "easy" problems with up to 230 jobs (Chu 1992a).

Since the problem discussed in this paper is NP-hard, only branch-and-bound approaches or dynamic programming approaches seem to be available to build exact methods. With unequal release dates, idle times may be inserted in the optimal schedules (Conway, Maxwell and Miller 1967). The presence of these idle times in optimal solutions destroys the usual scheme of dynamic programming approach. Therefore we use a branch-and-bound approach. For this purpose, more than dominance properties for general over-regular criteria, we prove additional dominance properties which are specific to the problem at hand.

3.1. Additional Dominance Properties

Theorem 8. If there are two jobs $j \in N\text{-}J(K)$ and $i \in J(K)$ in the kth position such that $F_i(t_i) \geq F_j(t_i)$ and $\{n - q(K) - 1\} \times (p_i - p_j) \leq Q_{i,j} - Q_{j,i}$, then $P(K,j)$ is dominated, where

$$Q_{u,v} = \max\left[F_u(t_i) - d_{[k]}, 0\right] + \max\left[F_v(E(K)) - d_{[q(K)+1]}, 0\right].$$

Proof. Consider the schedule $P(K,j)$. We construct another schedule S by interchanging the positions of jobs i and j (see Figure 7).

In the case where $p_i \leq p_j$, considering that $F_i(t_i) \geq F_j(t_i)$, it is clear that $C_i(S) \leq C_j(P(K,j))$ and $C_j(S) \leq C_i(P(K,j))$. The other jobs after job j can be scheduled earlier. The schedule $P(K,j)$ then is dominated.

In the case where $p_i > p_j$, the jobs after j in $P(K,j)$ are delayed in S (see Figure 7). This increases the completion time and consequently the tardiness of each of these jobs at most by $p_i - p_j$.The

number of these jobs is n-$q(K)$-1. On the other hand, the interchange of positions can reduce the tardiness of jobs i and j by $Q_{i,j}$-$Q_{j,i}$. From the assumption of Theorem 8, we have

$$Q(S)-Q(P(K,j))\le\{n-q(K)-1\}\times(p_i-p_j)-\{Q_{i,j}-Q_{j,i}\}\le 0.$$

As a consequence $P(K,j)$ is dominated.

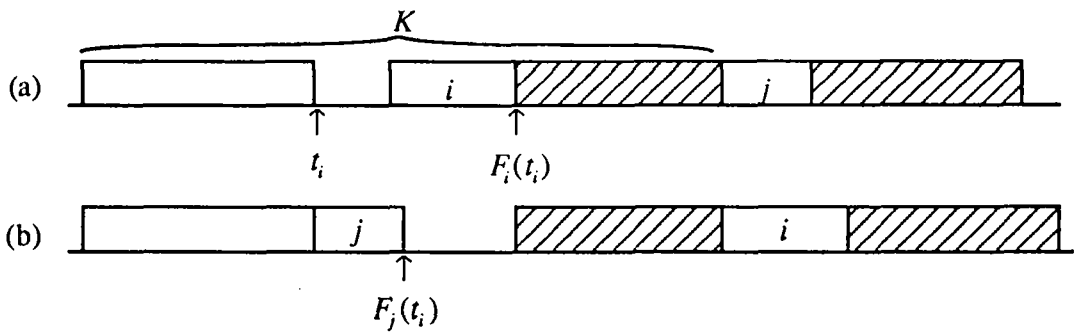

Figure 7. Interchange of positions of jobs i and j. (a) Schedule $P(K,j)$. (b) Schedule S.

Theorem 9. The schedule $P(K,i)$ is dominated if there is another partial solution K' such that $J(K')=J(K)\cup\{i\}$, $\{n-q(K')\}\times R_j(E(K'))+Q(K')\le\{n-q(K')\}\times R_j(E(K/i))+Q(K/i)$ and $Q(K')\le Q(K/i)$, where j is a job of N-$J(K')$ with the smallest release date.

The proof of Theorem 9 needs the following lemma that has to be proved first.

Lemma 1. Given a set W of jobs, if we construct two optimal schedules σ and σ' respectively starting from time t and t' ($t'\ge t$), then $Q(\sigma')$-$Q(\sigma)\le$card$(W)(t'-t)$.

Proof. We construct another schedule S starting from t' in which the jobs are scheduled in the same order as in σ, the completion time and consequently the tardiness of each job is increased at most by t'-t. We then have $Q(S)$-$Q(\sigma)\le$card$(W)(t'-t)$. In addition, from the definition of σ', we have $Q(\sigma')\le Q(S)$ which implies $Q(\sigma')$-$Q(\sigma)\le$card$(W)(t'-t)$.

Proof of Theorem 9. Consider the schedule $P(K,i)$ and the schedule S composed of K' and the optimal partial schedule of jobs N-$J(K')$, following K'. It is clear that in $P(K,i)$ the jobs N-$J(K')$ are optimally scheduled from time $R_j(E(K/i))$ and in S these jobs are optimally scheduled from time $R_j(E(K'))$.

If $R_j(E(K'))\leq R_j(E(K/i))$, it is obvious that $P(K,i)$ is dominated by S considering the assumption $Q(K')\leq Q(K/i)$.
If $R_j(E(K'))>R_j(E(K/i))$, the difference of total tardiness of the optimal partial schedules of jobs of N-$J(K')$ in S and $P(K,i)$ is at most $\{n-q(K')\}\times\{R_j(E(K'))-R_j(E(K/i))\}$ according to Lemma 1. This implies that

$$Q(S)-Q(P(K,i))\leq Q(K')-Q(K/i)+\{n-q(K')\}\times\{R_j(E(K'))-R_j(E(K/i))\}.$$

Considering the assumption of Theorem 9 we have $Q(S)\leq Q(P(K,i))$, which implies that the schedule $P(K,i)$ is dominated.

In a branch and bound algorithm, we also need a lower bound. In general, with unequal release dates and non pre-emptive jobs, the lower bound is obtained by relaxing the problem to a problem with pre-emptive jobs. We also use this scheme to compute lower bounds, because the problem with pre-emptive jobs, as shown in Section 2, is polynomially solvable.

Usually, a good initial solution can make a branch and bound algorithm more efficient. For this purpose, we use two heuristics called respectively EST (earliest start time) and ECT (earliest completion time) that we presented first.

3.2. Heuristic Algorithms

All the heuristics construct a new partial schedule K/i by adding an unscheduled job i behind the partial schedule K constructed before. K then becomes the new partial schedule. This continues until all the jobs are scheduled. Initially, the partial schedule K is set to Ø. We now explain the choice of a job i.

Earliest Start Time (EST) Priority Rule
Select job $i\in N$-$J(K)$ with the smallest earliest start time $R_i(E(K))\leq R_j(E(K))$ for all $j\in N$-$J(K)$. Break ties by choosing i with $\min(p_i)$, and further ties by choosing i with $\min(i)$. Schedule i and update K.

Earliest Completion Time (ECT) Priority Rule
Select job $i\in N$-$J(K)$ with the smallest earliest completion time $F_i(E(K))\leq F_j(E(K))$ for all $j\in N$-$J(K)$. Break ties by choosing i with $\min\{R_i(E(K))\}$, and further ties by choosing i with $\min(i)$. Schedule job i and update K.

It should be noticed that both of these heuristics can be efficiently implemented in $O(n\log n)$.

3.3. The Branch-and-Bound Algorithm

To construct an optimal schedule, we use a branch-and-bound technique. It uses a forward sequencing branching rule. Each node is defined by a partial sequence K of jobs scheduled from the beginning of the schedule. Initially, K is an empty set.

Each descendant node is obtained by adding, next to the partial sequence K, a new job i chosen among the unscheduled jobs. The choice of a job i is such that $P(K,i)$ is not dominated according to the dominance theorems presented in Section 2 and Subsection 3.1.

If the release dates of *all* the unscheduled jobs are smaller than the completion time of K, the remaining problem becomes equivalent to a problem with identical release dates and can be solved by ordering the unscheduled jobs in the SPT order.

The lower bound for each descendant node is computed by relaxing the sub-problem composed of unscheduled jobs into a problem with pre-emptive jobs and applying the SRPT rule to it. The initial feasible complete solution is obtained by taking the best one given by the EST and ECT heuristics. When a new feasible complete solution is obtained, and is better than the previous one, it is retained as the new best solution found so far. A descendant node is eliminated if its lower bound is greater than or equal to the total tardiness of the best solution found so far. We use a branch-and-bound with jumptracking, which means that we always select the node with the smallest lower bound for branching.

3.4. Computational Results

In order to evaluate the performance of the algorithm, we randomly generated a number of examples on which the algorithm was applied. The generation of examples was carried out based on uniform probability distributions. For the generation of processing times and release dates of jobs, we used the same scheme proposed by Hariri and Potts (1983) for the minimization of total weighted completion time. The processing times were generated between 1 and 100, and the release dates were generated between 0 and $50.5\times n\times\alpha$, α being a generation parameter. We propose to generate the generalized due dates in the following way. After having generated the release dates and processing times, we constructed a schedule σ by using the EST heuristic The generalized due dates were generated between 0 and $\beta\times C_{[n]}(\sigma)$, β being another generation parameter. The motivation to generate the generalized due dates in this way is the fact that $C_{[n]}(\sigma)$ is a lower bound of the makespan in all feasible schedules (Rinnooy Kan 1976). We carried out two series of experiments respectively with n=50 and n=70. For each series, the combination of 10 values of α (0.2, 0.4, 0.6, 0.8, 1.0, 1.25, 1.50, 1.75, 2.0,

3.0) and 4 values of β (0.5, 0.75, 1.0, 1.5) gave 40 sets of examples, each of which contained 20 randomly generated examples. In total 1,600 examples were generated.

Table I. Average Number of Nodes Considered (n=50)

	α									
β	0.2	0.4	0.6	0.8	1.0	1.25	1.50	1.75	2.0	3.0
0.5	107.30	443.85	1040.50	1537.90	1255.20	336.55	158.45	97.60	77.20	55.10
0.75	18.60	270.60	1767.15	1346.80	1069.30	258.10	222.65	113.50	83.20	57.95
1.0	36.40	13.90	1134.05	1226.90	1061.35	431.95	201.00	98.55	80.15	48.60
1.5	0.35	0.85	20.15	9.95	13.85	7.05	8.90	7.80	7.50	2.35

Table II. Average Computation Time (n=50)

	α									
β	0.2	0.4	0.6	0.8	1.0	1.25	1.50	1.75	2.0	3.0
0.5	7030	19649	26668	22871	11240	1996	663	358	267	175
0.75	1375	10426	43274	19245	9383	1327	1128	418	289	184
1.0	3142	506	55390	15600	8750	2642	774	356	274	161
1.5	51	58	621	193	154	76	80	72	68	43

Tables I and II respectively report the average number of nodes considered and the computation time consumed by the algorithm for the series with n=50.

Table III. Average Number of Nodes Considered (n=70)

	α									
β	0.2	0.4	0.6	0.8	1.0	1.25	1.50	1.75	2.0	3.0
0.5	377.20	1802.80	4636.60	4444.65	4405.75	1337.35	520.35	254.35	157.55	90.65
0.75	14.80	304.95	4947.30	4226.65	3133.10	863.65	380.60	523.30	164.95	91.65
1.0	3.15	24.40	188.65	2126.80	1895.55	1208.00	881.60	369.40	157.60	84.75
1.5	0.70	2.40	5.50	19.54	24.40	14.90	10.60	10.50	7.30	4.10

Table IV. Average Computation Time (n=70)

	α									
β	0.2	0.4	0.6	0.8	1.0	1.25	1.50	1.75	2.0	3.0
0.5	54238	136662	229638	119256	76905	15441	3754	1295	697	348
0.75	1418	24539	233284	124378	40434	7312	2313	4391	741	352
1.0	220	1299	6304	59359	18955	10471	5566	1941	689	333
1.5	86	138	195	463	350	186	126	119	97	74

Tables III and IV are analogue to Tables I and II but for the series with n=70. From the tables, we can see that problems are considerably less difficult for extreme values of α. In fact, when α is small, the problem easily becomes a problem without release dates after having scheduled a few jobs, which is polynomially solvable. When α is large, the number of active schedules to be considered is very small.

There is no simple relation between the hardiness of the problem and the variation of β, except when β is very large. In this case the problem becomes very easy because of the large due dates and the large number of feasible solutions whose total tardiness is zero. Among the tested examples, some were so simple that very few branchings took place (only a small number of nodes were considered). The most difficult case seems to be the one with α=0.6 and β=0.75. We can also remark that the problem difficulty increases very quickly with n.

When n=50 and α=0.6, the average number of nodes considered in the case β=0.75 is less than in the case β=0.50, but the mean computation time is larger. This is because there is an example in the case β=0.75 for which the search tree is very large, the algorithm had to take a lot of time to compare the lower bound of a new created node with the lower bound of existing nodes in order to keep the nodes in nondecreasing order of their lower bounds.

4. CONCLUDING REMARKS

In this paper, we defined a new class of criteria called over-regular criteria and established properties of these criteria. These properties are applicable for several scheduling problems and not specific for a particular problem as most work in the scheduling literature.

We also considered one of the over-regular criterion scheduling problem to minimize total tardiness with generalized due dates. This problem is relevant in real life, especially in maintenance departments where maintained parts are interchangeable. For this problem we proved supplementary dominance properties and proposed a branch-and-bound algorithm able to optimally solve problems with 70 jobs for the hardest case in our experiments. Further effort could be made to improve the efficiency of the algorithm and/or consider other over-regular criteria optimization.

REFERENCES

BAKER, K.R. 1974. *Introduction to Sequencing and Scheduling.* John Wiley & Sons, New York.

BAKER, K.R., AND J.W.M. BERTRAND. 1982. A Dynamic Priority Rule for Scheduling Against Due-Dates. *J. Opns. Man.* **3,** 37-42.

BAKER, K.R., AND L.E. SCHRAGE. 1978. Finding an Optimal Sequence by Dynamic Programming: An Extension to Precedence-Related Tasks. *Opns. Res.* **26,** 111-126.

CHU, C. 1992a. A Branch-and-Bound Algorithm to Minimize Total Tardiness with Different Release Dates. *Nav. Res. Logist.* **39,** 265-283.

CHU, C. 1992b. A Branch-and-Bound Algorithm to Minimize Total Flow Time with Unequal Release Dates. *Nav. Res. Logist.* **39/5,** to be published.

CHU, C., AND M.C. PORTMANN. 1992. Some New Efficient Methods to Solve the $n/1/r_i/\Sigma T_i$ Scheduling Problem. *Europ. J. Opnl. Res.* to be published.

CONWAY, R.W., W.L. MAXWELL AND L.W. MILLER. 1967. *Theory of Scheduling.* Addison-Wesley, Reading, MA.

DU, J., AND J.Y.-T. LEUNG. 1990. Minimizing Total Tardiness on One Machine is NP-hard. *Math. Opns. Res.* **15,** 483-495.

EMMONS, H. 1969. One-Machine Sequencing to Minimize Certain Functions of Job Tardiness. *Opns. Res.* **17,** 701-715.

FISHER, M.L. 1976. A Dual Algorithm for the One-Machine Scheduling Problem. *Math. Prog.* **11,** 229-251.

HALL, N.G. 1986. Scheduling Problems with Generalized Due Dates. *IIE Transactions* **18,** 220-222.

HALL, N.G., S. P. SETHI AND C. SRISKANDARAJAH. 1991. On the Complexity of Generalized Due Date Scheduling Problems. *Europ. J. Opnl. Res.* **51,** 100-109.

HARIRI, A.M.A., AND C.N. POTTS. 1983. An Algorithm for Single Machine Sequencing with Release Dates to Minimize Total Weighted Completion Time. *Discr. Appl. Math.* **5,** 99-109.

LAWLER, E.L. 1977. A "Pseudopolynomial" Algorithm for Sequencing Jobs to Minimize Total Tardiness. *Ann. Disc. Math.* **1,** 331-342.

LAWLER, E.L. 1982. A Fully Polynomial Approximation Scheme for the Total Tardiness Problem. *Opns. Res. Lett.* **1,** 207-208.

POTTS, C.N., AND L.N. VAN WASSENHOVE. 1982. A Decomposition Algorithm for the Single Machine Total Tardiness Problem. *Opns. Res. Lett.* **1,** 177-181.

RINNOOY KAN, A.H.G. 1976. *Machine Sequencing Problems: Classification, Complexity and Computation.* Nijhoff, The Hague.

SRINIVASAN, V. 1971. A Hybrid Algorithm for the One-Machine Sequencing Problem to Minimize Total Tardiness. *Nav. Res. Logist. Quart.* **18,** 317-327.

WILKERSON, L.J., AND J.D. IRWIN. 1971. An Improved Algorithm for Scheduling Independent Tasks. *AIIE Transactions* **3,** 239-245.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique.

ISSN 0249-6399